Chapter
Post-Quantum Cryptography: An Overview

Syed Ismail A[1], Priya Saxena[2], Priyanshi Singh[3]
[1]SRM Institute of Science and Technology, Chennai,
syedisma@srmist.edu.in
[2]SRM Institute of Science and Technology, Chennai,
ps0751@srmist.edu.in,
[3]SRM Institute of Science and Technology, Chennai,
ps5913@srmist.edu.in

1 Introduction

Quantum computing promises unprecedented computational power, but also poses a grave threat to our current encryption methods[3, 36]. This chapter explains the nature of this threat, the urgency of transitioning to post-quantum cryptography (PQC), and the global efforts underway to secure our digital future[12, 21] . We analyze the quantum vulnerability of classical cryptography, including RSA and elliptic curve systems, and explore leading PQC alternatives including lattice-based cryptography (CRYSTALS-Kyber and Dilithium), multivariate cryptography (Rainbow and GeMSS), and isogeny-based approaches. We also examine real-world applications in blockchain, secure communications, IoT, and government adoption, highlighting implementation challenges and future research directions. While quantum computers powerful enough to break current encryption are still under development, the time to act is now to prevent future data breaches[10, 44] .

1.1 The Quantum Computing Threat to Classical Cryptography

Quantum computing is a paradigm shift in computing power, based on principles of quantum mechanics—specifically superposition and entanglement—to execute certain computations exponentially much faster than classical computers[6, 15]. In con-

Syed Ismail A
SRM Institute of Science and Technology, Chennai, India e-mail: syedisma@srmist.edu.in

Priya Saxena
SRM Institute of Science and Technology, Chennai, India e-mail: ps0751@srmist.edu.in

Priyanshi Singh
SRM Institute of Science and Technology, Chennai, India e-mail: priyanshisingh.10009@gmail.com

trast to classical bits, which are in a deterministic state of 0 or 1, quantum bits (qubits) can be in a superposition of both states, facilitating enormous parallelism. Entanglement also makes it possible for qubits to have correlated states, facilitating highly efficient information processing that is not possible with classical architectures[6, 15].

How Quantum Computers Break Encryption

Our existing encryption mechanisms, which safeguard everything from financial transactions to national secrets, rely on mathematical problems that are computationally expensive for classical computers[14, 36] . The most troubling of these are:

- *RSA (Rivest-Shamir-Adleman)*[14] : Is predicated on the difficulty of factoring large composite numbers into primes. The larger the number, the longer classically it will take, so brute force is not practical until quantum computers.
- *ECC (Elliptic Curve Cryptography)*[7] : Tend to be used to secure mobile phones and web sites, ECC depends on the difficulty of cracking the elliptic curve discrete logarithm problem (ECDLP)[7, 15].

Both of these crypto primitives depend on asymmetry—mathematical challenges that are computationally simple to check but exponentially difficult to solve. *Quantum computers destroy this asymmetry.*

Shor's Algorithm and its Implications

The algorithm behind this threat is Shor's algorithm[15], developed by *Peter Shor* in 1994. In contrast to traditional factorization algorithms, which take sub-exponential time, Shor's algorithm takes polynomial time. That is, when the number to be factored is larger, the computation time increases much more slowly on a quantum computer than on a classical computer.

1. **Reduction to Order-Finding:**
   The security of RSA[14] relies on the fact that given a large integer $N = pq$ (product of two primes), factoring N is computationally infeasible. Nevertheless, this issue can be mapped to determining the order of an arbitrarily selected integer a, defined as the smallest r such that:

$$a^r = 1 \ (mod \ N) \tag{.1}$$

2. **Quantum Fourier Transform (QFT):**
   Classical computers struggle to compute this order efficiently. However, quantum computers leverage the Quantum Fourier Transform [6] (QFT) to effi-

ciently determine the period r, leading to factorization in polynomial time $O\left((\log\log N)^3\right)$

What This Means: A quantum computer powerful enough to run Shor's algorithm could undermine the security of RSA[14] and ECC[7] and potentially leak sensitive information across the globe. Although present-day quantum computers are not yet ready to run Shor's algorithm at a useful scale because of qubit coherence and error correction limitations, specialists predict that large-scale fault-tolerant quantum computers will be a reality within the next 10-15 years. This time frame is important because cryptographic transitions are years, if not decades.

Current Research Outlook:

While current quantum computers lack sufficient qubit coherence and error correction to run Shor's algorithm at practical scales, estimates suggest that within the next 15–20 years[10, 14], large-scale, fault-tolerant quantum computers could become a reality. Given that cryptographic transitions take decades, this presents an urgent threat.

1.2 The Urgency of Post-Quantum Cryptography (PQC)

The existential threat of quantum computing is not limited to a distant future scenario; it is a present concern due to "harvest now, decrypt later" attacks[10].

The "Harvest Now, Decrypt Later" Problem

Adversaries today are actively storing encrypted communications with the expectation that they will be decrypted once quantum computers become viable. This is particularly concerning for:

- *Government Information:* Spy agencies can intercept encrypted messages today and decrypt them at a later date, creating grave national security threats[13].
- *Medical and Financial Records:* Sensitive information has to be kept confidential for decades. A future breach could reveal long-term personal or financial data[3].
- *Blockchain and Cryptographic Signature:* Transactions that are secured through RSA[14]- or ECC[7]-based cryptography can be susceptible to retroactive attacks[34, 40], which can compromise trust in digital signatures.

Case Study: The NSA's Quantum Cryptography Initiative

The National Security Agency (NSA) has already deprecated Suite B cryptography standards (which were based on ECC[7]) as a result of the threat from quantum

computing (NSA, 2015). This clearly shows that the intelligence agencies are extremely serious about the quantum threat, adding impetus to transition[13].

The Complexities of Migrating to Post-Quantum Cryptography

Moving to post-quantum cryptography (PQC) is a difficult, expensive, and time-consuming process. Even if quantum computers are ten years away, the migration to PQC needs to start now.

1. Research and Standardization: Organizations like NIST (National Institute of Standards and Technology[2, 41]) initiated the Post-Quantum Cryptography[3] Standardization Project in 2016 to assess algorithms that are both classical and quantum attack-resistant. Algorithms under consideration include lattice-based cryptography (e.g., CRYSTALS-Kyber[25]), hash-based cryptography, and multivariate-quadratic equations[12, 21].
2. Implementation Complexity: PQC involves updating protocols, software libraries, network security stacks, and hardware—a gigantic worldwide infrastructure transformation. Most currently installed embedded systems [45] (IoT devices, secure chips, industrial control systems) can't be upgraded easily and might need completely new cryptographic hardware[12, 44].
3. Economic and Industrial Impact: Telecommunications, finance, and government infrastructure are greatly dependent on RSA and ECC. Putting the transition off might result in devastating security breakdowns when quantum computers become viable, costing possibly billions of dollars and undermining the trust of the public.

1.3 Global Efforts Toward Quantum-Secure Cryptography

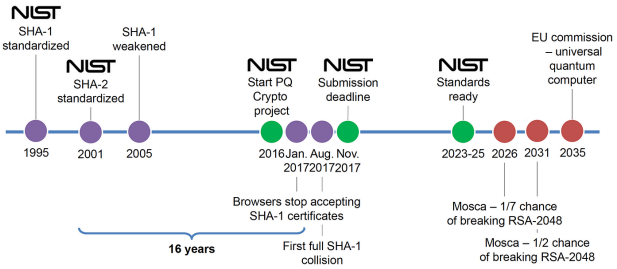

Fig. .1 Timeline of Cryptographic Standards and the Transition to Post-Quantum Cryptography

This timeline (Figure 1.1) presents a historical overview of cryptographic standards development, alongside projections for quantum computing capabilities. Several key phases and initiatives are highlighted. The left side of the timeline (purple and blue dots) depicts the evolution of classical cryptographic standards, such as SHA-1 and SHA-2 [34]. We see the standardization of SHA-1 in 1995, followed by its eventual weakening in 2005, and the subsequent standardization of SHA-2 in 2001. This illustrates the iterative nature of cryptography: as attacks improve, new standards must be developed. The bracket indicating "16 years" shows the time elapsed between SHA-1 standardization, and the first full SHA-1 collision.

The central and right-hand portions of the timeline (green and red dots) focus on the response to the quantum threat. The National Institute of Standards and Technology (NIST) plays a central role, shown by the green dots. NIST[41] initiated its Post-Quantum Cryptography (PQC) project in 2016, with a submission deadline in 2017, and an anticipated publication of final standards around 2023-25. The red dots represent projections related to quantum computing capabilities, including Mosca's estimates for the probability of breaking RSA-2048, and the EU Commission's projection for a "universal quantum computer." These global efforts are further categorized as follows:

NIST's PQC Standardization Initiative (2016-Present):

- The NIST competition has identified four primary algorithms for standardization (e.g., CRYSTALS-Kyber[25] for key exchange[4]).
- Expected publication of final standards in 2024.

European & Chinese Initiatives:

- The EU's OpenQKD Project and China's Quantum Satellite (Micius) demonstrate major investments in quantum-safe communications.
- China is leading in quantum key distribution (QKD), which offers quantum-resistant encryption via entanglement-based key exchange[4].

Commercial Adoption:

- IBM, Google, and Microsoft are already integrating PQC into their cloud security infrastructures[44].
- Tech giants are investing heavily in hybrid cryptographic models, blending PQC with classical encryption for a smooth transition.

Key Steps Forward

- *Accelerate PQC adoption:* Governments, corporations, and standards bodies must work together to finalize and implement PQC standards as quickly as possible.

- *Invest in cryptographic agility:* Organizations must design systems that can easily switch between cryptographic algorithms to adapt to future threats.
- *Monitor advancements in quantum computing hardware:* Staying informed about the latest developments in quantum computing is essential for anticipating timelines and adapting security strategies.

The cryptographic infrastructure of the future must be built today—before it's too late. The secrets we seek to protect tomorrow demand the action we take today.

2 Literature Review

The Looming Quantum Threat and the Rise of PQC

Picture a future where all of our secure messages, from state secrets to banking transactions, become susceptible to being eavesdropped upon. This is the future that lies ahead with the development of quantum computers. Post-Quantum Cryptography[3] (PQC) is our guarantee against this possibility, which can resist the attacks of both classical and quantum computers. PQC is significant due to the potential of quantum computers to perform mathematical problems underlying current cryptographic techniques efficiently.

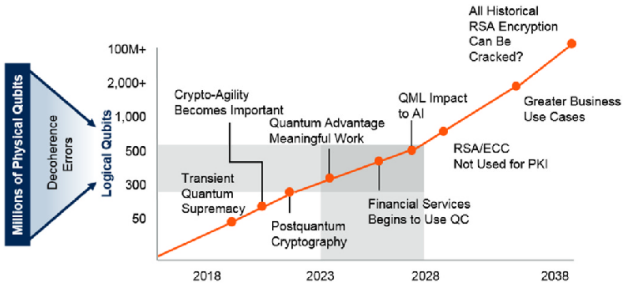

Fig. .2 Projected Timeline of Quantum Computing Capabilities and its Impact on Cryptographic Security

The timeline in Figure 1.2 projects the increase in logical qubits, a key metric for quantum computing power. It highlights milestones related to the transition to post-quantum cryptography (PQC), the phasing out of vulnerable algorithms like RSA[14] and ECC[7], and the eventual capability of quantum computers to break existing encryption. The shaded areas indicate phases such as "Transient Quantum Supremacy" and "Quantum Advantage," where quantum computers begin to outperform classical computers in specific tasks. The y-axis on the left shows number of Physical Qubits, and on the right are Logical qubits. Shor's Algorithm, a quantum algorithm for efficient factorization of large numbers and computation of

discrete logarithms in polynomial time, is most seriously under threat. This aspect is based on numerous ubiquitous cryptographic protocols such as RSA and ECC that rely on these problems' assumed intractability.

2.1 How Shor's Algorithm Breaks RSA and ECC

Shor's Algorithm takes advantage of quantum entanglement and superposition to perform computations that would be impractical for classical computers. The mathematical insight is that it can take advantage of the Quantum Fourier Transform (QFT)[6] to determine the periodicity of a modular exponentiation function—a computation that is difficult for classical computers.

To see the significance of Shor's algorithm, it's useful to think of the following:

- *The Classical Bottleneck:* Conventional cryptographic systems are based on mathematical problems that can be easily executed in one direction but are prohibitively hard to reverse without having the proper "key." Multiplying two prime numbers, say, is simple, but determining those primes from the product is exponentially more difficult for classical computers as the primes increase.
- *The Quantum Advantage:* Shor's algorithm avoids this problem by using quantum superposition and entanglement to try many possible solutions at once. The QFT serves as a form of "quantum signal processing" tool, detecting the periodic internal structure of the mathematical problem, exposing hidden factors.
  - *QFT Insight:* The Quantum Fourier Transform[6] efficiently detects hidden patterns, much like how a prism separates white light into its constituent colors.

Shor's algorithm operates by:

1. Selecting a random integer coprime to N.
2. Computing the order r, which is the smallest integer such that $a^r \equiv 1 \; mod \; N$ (a critical step that relates directly to factoring the integer N).
3. Exploiting quantum phase estimation and QFT to determine r, which then allows factoring N efficiently via the continued fraction method.

Grover's Algorithm and Its Impact on Symmetric Cryptography

While Shor's Algorithm threatens the foundations of public-key cryptography, Grover's Algorithm[6] undermines symmetric-key encryption by significantly accelerating brute-force attacks. Additionally, *Grover's Algorithm* accelerates brute-force search attacks by reducing search complexity from $O(N)$ to $O(\sqrt{N})$, effectively cutting the security of symmetric cryptography in half.

For example:

- AES-128 security is reduced to AES-64, making brute-force attacks feasible[34].
- SHA-256 hash security is weakened to the level of SHA-128, threatening blockchain integrity and digital signatures.

Thus, while doubling key lengths in symmetric cryptography mitigates this issue, it is not a long-term solution. A quantum-resistant approach must be adopted across cryptographic systems. While Grover's quadratic speedup is significant, it is nowhere near as devastating as Shor's exponential advantage in breaking asymmetric cryptography.

2.2 What Makes a Cryptosystem Quantum-Resistant?

A cryptosystem is said to be quantum-resistant if it is built on problems that are not efficiently solvable by any known quantum algorithms, i.e., they are not in *BQP (Bounded-Error Quantum Polynomial Time)*—the class of decision problems that can be decided by a quantum computer in polynomial time with high probability of correctness.

In addition to withstanding quantum attacks on asymmetric cryptography, it must:

- *Resist Grover's Algorithm*[6], ensuring that symmetric encryption remains secure.
- Exhibit worst-case to average-case hardness guarantees, meaning security holds even in the most challenging conditions, not just on average[17].

Comparison of Classical vs. Post-Quantum Cryptographic Assumptions

The table below highlights the differences between classical and post-quantum cryptographic assumptions:

Table .1 Classical vs. Post-Quantum Cryptographic Assumptions

Problem	**Classical Complexity**	**Quantum Complexity**
Integer Factorization (RSA)	Sub-exponential $O\left(e^{(\log\log N)^{1/\kappa}}(\log\log\log N)^{2/\kappa}\right)$	Polynomial $O\left((\log\log N)^3\right)$
Discrete Logarithm (ECC)	Sub-exponential	Polynomial $O\left((\log\log N)^3\right)$
Symmetric Key Search	$O(N)$	$O\left(\sqrt{N}\right)$ (Grover's Algorithm)
Lattice-Based (LWE, SIS)	Exponential $2^{O(n)}$	Exponential $2^{O(n)}$
Code-Based (McEliece)	Exponential	Exponential
Multivariate Equations (MQ)	NP-hard	NP-hard

.

3 Methodology

The prospect of fault-tolerant quantum computers presents a serious challenge to contemporary cryptographic practice. Although larger key sizes in symmetric-key ciphers offer some increased protection, the intrinsic weakness of extensively deployed public-key algorithms, like RSA and those based on elliptic curves, necessitates fundamentally new cryptographic approaches. This section introduces several key families of post-quantum cryptographic (PQC) schemes – specifically lattice-based, hash-based, multivariate, code-based, and isogeny-based constructions. Each of these families represents a distinct paradigm, offering varying trade-offs in terms of provable security, computational efficiency, memory footprint, and suitability for different deployment scenarios. A thorough understanding of these diverse approaches is essential for navigating the changing landscape of PQC and making informed decisions about the long-term security of digital systems.

3.1 Lattice-Based Cryptography

Lattice-based cryptography[22] provides a novel and promising method for constructing cryptographic systems that are secure against quantum computer attacks. Unlike conventional methods based on integer factorization or discrete logarithms, lattice-based cryptography finds its security in the internal mathematical properties of lattices – ordered collections of points in multi-dimensional space. The construction essentially hinges upon the challenge in solving specific geometric problems in double the lattices, challenges considered to stay challenging even to quantum computers.

What is a lattice?

Essentially, a lattice is a discrete n-dimensional subgroup of Euclidean space ($\mathbb{R}^n$). To get a feel for what a lattice is, imagine an infinitely repeating array of points in space. More rigorously, a lattice L can be defined as the set of all integer linear combinations of a set of n linearly independent vectors $b_1, \ldots, b_n$ in $\mathbb{R}^n$.

They create a basis for the lattice:

$$L\left(b_1, \ldots, b_n\right) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \right\} \tag{.2}$$

Despite its abstract nature, this mathematical structure underpins a robust class of cryptographic systems by providing unique security properties. Lattices are fundamental objects in mathematics, and they are characterized by their structure. However, what makes them interesting in cryptography is that certain problems involving

lattices are believed to be computationally *hard*, even for quantum computers. This hardness provides a foundation for building secure cryptographic schemes.

A key characteristic of lattices is that they can be defined by multiple bases. While a lattice is uniquely defined as a set of points, the particular set of basis vectors used to represent it can vary greatly. Some bases consist of short, nearly orthogonal vectors, making it easy to "see" the structure of the lattice and solve certain problems. These are considered "good" bases. In contrast, other bases consist of long, nearly parallel vectors, obscuring the structure of the lattice and making it difficult to find short vectors within it. These are considered "bad" bases.

A basis of L is an ordered set $B = (b_1, b_2, \ldots, b_n)$ such that

$$L = L(B) = B \cdot \mathbb{Z}^n = \left\{ \sum_{i=1}^{n} c_i b_i : c_i \in \mathbb{Z} \right\} \tag{.3}$$

Note that by convention, b_i are column vectors and

$$B \cdot k = k_1 b_1 + \ldots + k_n b_n \tag{.4}$$

where k is a column vector.

The fundamental parallelepiped of basis B is

$$P(B) = B \cdot \left[-\frac{1}{2}, \frac{1}{2} \right]^n \tag{.5}$$

$$P(B) = \left\{ \sum_{i=1}^{n} \alpha_i b_i : -\frac{1}{2} \leq \alpha_i < \frac{1}{2} \right\} \tag{.6}$$

Observe that $P(B)$ not only depends upon the lattice, but also upon the selection of basis B. A "good" basis of L yields a fairly square-like parallelepiped, but a 'bad' basis yields a very narrow parallelepiped. It is easy to verify the following lemma.

The key to lattice cryptography lies in leveraging the contrast between well-behaved and poorly-behaved bases for security purposes. The heart of many lattice-based cryptosystems lies in the difficulty of transforming a bad basis into a good basis. A cryptosystem might use a good basis as a private key and a bad basis as a public key, effectively hiding the underlying lattice structure.

3.2 Fundamental Hard Problems: SVP and CVP

This difficulty in transforming between bases is closely tied to challenging computational problems, such as the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP).

1. *Shortest Vector Problem (SVP):* Given a basis for a lattice, the Shortest Vector Problem (SVP) asks to find the shortest non-zero vector in the lattice. Intuitively, this is like finding the smallest possible step you can take to move from the

origin to another point in the lattice. Formally, for a lattice L generated by a basis $B = \{b_1, b_2, \ldots, b_n\}$, the SVP seeks a vector $v \in L \setminus \{0\}$ such that:

$$|v| \leq |u| \quad \text{for all } u \in L \setminus \{0\} \tag{.7}$$

2. *Closest Vector Problem (CVP):* Given a basis for a lattice and a target point (not necessarily on the lattice), the Closest Vector Problem (CVP) asks to find the lattice point closest to the target point. To see a sense of what a lattice feels like, suppose an infinite repetitive pattern of points in space. This is analogous to identifying the closest "house address" to a certain location. Formally, for a lattice L and a target vector $t \in \mathbb{R}^n$, the CVP seeks a vector $v \in L$ such that:

$$|v - t| \leq |u - t| \quad \text{for all } u \in L \tag{.8}$$

Both SVP and CVP are known to be *NP-hard*, meaning that no efficient algorithm is known to solve them exactly in the worst case, and they are believed to be computationally intractable for high-dimensional lattices. However, their approximate versions—where one seeks a vector that is close to the shortest or closest, within a certain factor—are also believed to be hard. In the approximate versions, the goal is to find a vector whose length or distance is within a multiplicative factor (e.g., γ) of the optimal solution, where γ typically depends on the lattice dimension n.

The hardness of such approximate problems plays a central role in lattice-based cryptography. Cryptographic constructions frequently use the hardness assumption that approximatesolving such approximategoals, particularly in the worst-case, is computationally difficult. This worst-case hardness constitutes a powerful theoretical basis for the security of lattice-based cryptosystems.

From Hard Problems to Cryptographic Primitives[16]

The hardness of SVP and CVP, and especially their approximate variants, is the backbone for building a variety of cryptographic primitives. Some of these include hash functions, public key encryption schemes, digital signatures, and key exchange protocols. The move from abstract lattice problems [16][23] to real-world cryptographic tools is all about designing schemes carefully so that they use the hardness of these problems efficiently and in an usable manner.

For example, collision-resistant hash functions can be constructed from lattices using the hardness of finding short vectors in specifically designed lattices. In the same vein, public-key encryption schemes depend on the concept of having a "good" basis (private key) and a "bad" basis (public key) for a lattice, where encryption adds noise that can only be eliminated with the private key.

Building Blocks: Hash Functions and Public-Key Encryption

1. *Hash Functions:*
   - A hash function[32] h takes an input x (of any size) to an output $h(x)$ (of fixed size), such that it is computationally infeasible to discover two distinct inputs x_1 and x_2 that hash to the same output (i.e., $h(x_1) = h(x_2)$. Lattice-based hash functions, such as the *SWIFFT* hash function, take advantage of the fact that collisions are found by finding short vectors in carefully designed lattices. SWIFFT, which uses *Fast Fourier Transforms (FFTs)* over rings, offers both efficiency and a provably secure connection to lattice hardness.
2. *Public-Key Encryption:*
   - The distinction between "good" and "bad" bases, as established in the previous discussion of lattice properties, forms the foundation of lattice-based public-key encryption schemes. This fundamental characteristic of lattices enables the construction of secure public-key systems. The public key comprises a "bad" basis of the lattice, while the private key consists of the corresponding "good" basis. The encryption process involves the addition of a small noise vector to a lattice point. The decryption process, feasible only with knowledge of the "good" basis, enables the efficient removal of this noise to recover the original message. Decryption uses the private key to recover the original message by removing the noise.

Applications and Implementations

Lattice-based cryptography has real-world potential. Its applications span a wide range of domains, including:

- *Secure Communication:* Lattice-based key exchange mechanisms can replace vulnerable protocols in TLS/SSL for secure web communication.
- *Digital Signatures:* Lattice-based signature schemes can be used to authenticate transactions and secure digital identities.
- *Financial sector and Government:* Lattice-based methods can be used in payment systems, banks, IoT, etc to secure transactions and secure classified data.

Beyond these general applications, there are specific implementations in play. An ongoing transition includes adopting NIST, in which industry is advancing PQC standardization.

3.3 CRYSTALS-Kyber: A Module-LWE Key Encapsulation Mechanism for the Quantum Age[43]

CRYSTALS-Kyber, a lattice-based key encapsulation mechanism (KEM)[43], was standardized by NIST in 2022 as part of its post-quantum cryptography (PQC) project[25]. As a quantum and classical attack-resistant design, Kyber uses the hardness of the Module Learning With Errors (MLWE) problem to protect key exchange protocols. Its efficiency, configurability, and strong security proofs make it a pillar of quantum-resistant cryptographic infrastructure.

From LWE[26] to Module-LWE

Kyber's security is based on the Module Learning With Errors (MLWE) problem, a structured variant of the Learning With Errors (LWE) problem tailored for practical efficiency[24]. In contrast to unstructured LWE, MLWE works over lattices built from modules—algebraic objects that combine vectors and polynomial rings. Precisely, for a module rank k, the MLWE problem is defined as follows:

Given a matrix $A \in R_q^{k\times k}$, (with entries from a polynomial ring $R_q = \mathbb{Z}_q[x]/(x^n+1)$), a secret vector $s \in R_q^k$, and an error vector $e \in R_q^k$ from a small Gaussian distribution, distinguishing the pair $(A, As+e)$ from uniformly random (A, u) is computationally infeasible. This form decreases key sizes and speeds up computations while maintaining worst-case hardness guarantees.

Table .2 Kyber Security Levels

Parameter Set	Security Level (NIST)	Analogous AES Security
Kyber512	Level 1	AES-128
Kyber768	Level 3	AES-192
Kyber1024	Level 5	AES-256

Three levels of security are supported by Kyber through tunable module ranks k, as shown in Table .2. The parameter k controls the dimension of the lattice directly, making computational overhead and increased security trade off with each other.

3.4 Key Generation, Encapsulation, and Decapsulation: The Inner Workings of Kyber

Kyber's functionality can be decomposed into three major steps: key generation, encapsulation, and decapsulation. enumitem

A. Key Generation:

1. The key generation process begins with sampling matrix A, which is publicly available but does not leak any secret information.
2. Next, a secret key s and error vector e are produced by the code and chosen from certain probability distributions. These distributions play a critical role in the security of Kyber.
3. The public key b is generated from the matrix-vector product, and this helps the code make the final public key equal to $(A, b = As + e)$.

B. Encapsulation:

1. Alice, who'd like to send Bob a secret key, begins with sampling matrix x (the noise that helps generate a ciphertext) to create a public key, so that the recipient can decapsulate the code.
2. From here, Alice uses Bob's public key b to generate values. After creating key k, using matrix A, key x, and message m, the resulting public key is generated

$$ct = (At + e, b + e + m) \tag{.9}$$

C. Decapsulation:

1. Bob receives the ciphertext ct and decapsulates this code using the private key (s, e).
2. After this code is complete, Bob can generate the key with the calculations that $m \approx c - sx'$, allowing Bob to get the key k.

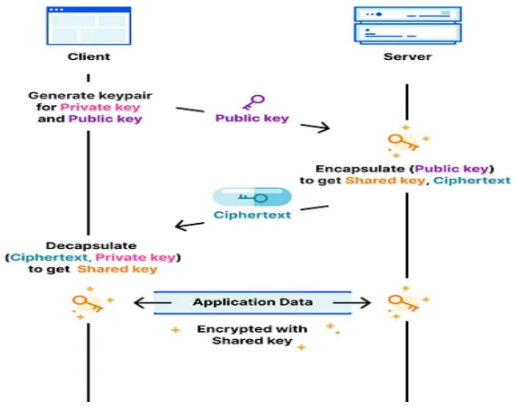

Fig. .3 Key Encapsulation Mechanism(KEM)

Figure 1.3 illustrates the Kyber key encapsulation mechanism (KEM) for secure communication. The client generates a key pair and shares the public key with the server, which encapsulates a shared key and ciphertext. The client then decapsulates the ciphertext to retrieve the shared key, enabling encrypted data exchange. Kyber's efficiency, strong security based on the Module-LWE problem, and NIST standardization make it a leading post-quantum cryptographic solution.

Kyber enjoys several significant advantages:

1. *Efficiency:* Kyber demonstrates excellent performance, particularly in key generation and encapsulation. Its optimized implementations make it suitable for real-world applications, including resource-constrained environments.
2. *Strong Security Proofs:* The security of Kyber relies on the Module-LWE problem, which has been thoroughly researched and is conjectured to be hard for both quantum and classical computers. Additionally, it comes with strong security proofs that instill trust in its resilience against known attacks.
3. *NIST Standardization[2]:* Kyber's choice as a standardized KEM[43] by NIST indicates its maturity and readiness for mass adoption. Standardization enables interoperability and inspires trust in its security. Standardization makes it accessible to all.
4. *Superior:* Many people currently think that Kyber is the superior code.

Although it has numerous advantages, Kyber also exhibits some disadvantages:

1. *Large Key Sizes:* In comparison to classical elliptic curve-based schemes, Kyber's key sizes are much larger. Although not impossible, the larger key sizes can affect storage requirements and bandwidth usage, particularly in environments that are resource-scarce.
2. *Hardware Optimization:* Efficient implementations are possible, yet hardware optimization of Kyber, especially on embedded platforms, is an area of active research. Customizing Kyber implementations for targeted hardware platforms can continue to optimize performance and save power.
3. *Transition Efforts:* The transition is still ongoing, wherein a hybrid mode needs to be defined between quantum and classical computing forms to assist with optimizing compatibility.

Kyber is a major breakthrough in post-quantum cryptography, a practical and secure key encapsulation solution. Its adoption as a NIST standard and positive performance properties make it a foundation of quantum-resistant infrastructure going forward.

3.5 CRYSTALS-Dilithium[27]: A Post-Quantum Signature Scheme for a Secure Future

Kyber fills the gap in quantum-resistant key exchange, yet secure communication also needs strong authentication techniques, CRYSTALS-Dilithium, a companion scheme in the CRYSTALS suite and a NIST-standardized post-quantum digital signature algorithm, fills this critical gap. Like Kyber, Dilithium is built upon hard problems in lattice theory, specifically Module-LWE (MLWE). Dilithium creates the encryption through rings, making more structure with it and also reducing complexity. The security rests on the difficulty of solving these ring/module-LWE, and more information about it is on the Kyber[27]. By having these features, the efficiency gets ramped up and also is adaptable on all devices with it.

Dilithium's Signature Generation Process

Dilithium's signature generation process involves several key steps:

1. **Key Generation:** The process begins by establishing a public-secret key pair rooted in the hardness of the Module Learning With Errors (MLWE) problem. A uniformly random matrix $A \in R_q^{k \times l}$ is sampled, where $R_q = \mathbb{Z}_q[x]/(x^{256}+1)$ is a polynomial ring with modulus q=8,329. Two short secret vectors $s_1 \in R_q^l$ and $s_2 \in R_q^k$, are generated from a bounded Gaussian distribution η. The public key is derived by computing the noisy linear combination $t = As_1 + s_2 \mod q$ which binds the public key (A, t) to the secrets (s_1, s_2). This ensures secrecy under the MLWE assumption, as recovering s_1 or s_2 from t requires solving a computationally infeasible lattice problem.
2. **Commitment Phase:** To initiate the signing of a message m, the signer first prepares a masked commitment to obscure secret-dependent values. A short random masking vector $y \in R_q^l$ is sampled, and the commitment $w = Ay \mod q$ is computed. This step conceals y while binding it to the public matrix A. Polynomial multiplications in R_q are accelerated using the Number Theoretic Transform (NTT), a technique analogous to the Fast Fourier Transform, which reduces computational complexity from $O(n^2)$ to $O(n \log n)$.
3. **Challenge Generation (Fiat-Shamir Heuristic):** The Fiat-Shamir heuristic replaces interactive protocols with a non-interactive process by deriving a deterministic challenge c from the message m and the commitment w. The challenge is generated as $c = H(m||\text{HighBits}(w))$ where H is a cryptographic hash function (e.g., SHA3-512). By binding c to both m and w, this step ensures non-malleability and thwarts replay attacks, as altering either the message or the commitment invalidates the challenge.
4. **Response with Aborts:** The signer computes the response vector $z = y + cs_1$, which embeds the secret s_1 into the masked vector y. To prevent leakage of secret key information, the protocol employs rejection sampling: if the infinity norm

$\| z \|_\infty$ exceeds a pre-defined bound η, the signing process aborts and restarts. This ensures the distribution of signatures remains statistically independent of $w - cs$, even under adaptive attacks.

5. **Hint Generation:** To enable efficient verification, a hint h is generated using $h = \text{MakeHint}(-s_2, w - cs_2 \mod q)$. This hint compresses the low-order bits of the error term $w - cs_2$, allowing verifiers to correct for noise without direct knowledge of s_2. By omitting full error terms, the hint minimizes signature size while preserving correctness.
6. **Signature Output:** The final signature is the compact tuple $\sigma = (z, c, h)$ with a size of approximately 2–4 KB. This balances security and practicality, making Dilithium suitable for resource-constrained environments like IoT devices. The components include z : A bounded response vector derived from the masked secret, c: a challenge deterministically tied to the message and commitment and h: hint for error correction during verification.

The Need for Post-Quantum Signatures: Why Dilithium?

Elliptic Curve Digital Signature Algorithm (ECDSA) is the current industry standard for classical digital signatures, securing protocols like TLS/SSL, blockchain (e.g., Bitcoin), and digital certificates. However, ECDSA relies on the elliptic curve discrete logarithm problem (ECDLP) [7,15], which is vulnerable to Shor's algorithm—a quantum algorithm that efficiently solves ECDLP, rendering ECDSA insecure in the quantum era. CRYSTALS-Dilithium, in contrast, is designed to resist quantum attacks by basing its security on lattice problems (Module-LWE and Module-SIS) that are currently intractable for both classical and quantum computers. Below, we compare the two schemes:

Table .3 ECDSA vs. CRYSTALS-Dilithium

Feature	**ECDSA**	**CRYSTALS-Dilithium**
Security Basis	Elliptic Curve Discrete Logarithm Problem	Module-LWE and Module-SIS lattice problems
Quantum Resistance	Broken by Shor's algorithm	Resistant to known quantum attacks
Public Key Size	32 bytes (ECDSA-256)	1,312 bytes (Dilithium2)
Signature Size	64 bytes (ECDSA-256)	2,420 bytes (Dilithium2)
Signing Speed	~0.1 ms (modern CPU)	~1.3 ms (modern CPU)
Verification Speed	~0.05 ms (modern CPU)	~0.2 ms (modern CPU)
Standardization	NIST FIPS 186-5 (classical)	NIST PQC Standard (post-quantum)

ECDSA's reliance on *ECDLP*[7,15] makes it insecure in a quantum future. *Dilithium's lattice-based security provides a quantum-resistant alternative.* While ECDSA outperforms Dilithium in speed and compactness (due to smaller keys/signatures), these advantages become irrelevant if quantum computers break ECDSA. ECDSA remains standardized for classical use, but Dilithium is part of NIST's Post-Quantum Cryptography (PQC) standardization process, ensuring its readiness for quantum threats.

While Dilithium represents a significant leap toward quantum-resistant cryptography, it is not without challenges:

1. **Larger Key and Signature Sizes:** Dilithium's public keys (≈1–2 KB) and signatures (≈2–4 KB) are orders of magnitude larger than ECDSA's (32-byte keys, 64-byte signatures). This imposes bandwidth and storage overheads, particularly in resource-constrained environments like IoT devices or blockchain networks.
2. **Computational Overhead:** Despite optimizations (e.g., NTT acceleration), Dilithium's signing and verification times (≈1–2 ms) lag behind ECDSA's (≈0.1 ms). While manageable for most applications, this gap matters in ultra-low-latency systems like high-frequency trading.
3. **Standardization and Interoperability:** As a NIST finalist (not yet fully standardized for all use cases), Dilithium requires broader ecosystem support. Legacy systems relying on ECDSA/RSA face integration hurdles during the transition.
4. **Side-Channel Resistance:** Like most lattice-based schemes, Dilithium's reliance on Gaussian sampling and rejection sampling raises concerns about side-channel attacks. Hardware instantiations need to have countermeasures, which raise design complexity.

Dilithium's constraints are indicative of the infancy of post-quantum cryptography rather than essential weaknesses. Its increased keys and reduced speeds are realistic compromises on quantum resistance—a forward-thinking investment in future-proofed security. While research works through these hurdles, Dilithium is likely to be the foundation of safe authentication in the quantum age, just as AES became the basis for secure authentication in the 2000s.

3.6 Multivariate Cryptography[28]

While lattice-based cryptography takes the lead in post-quantum due to its strong reductions, multivariate cryptography offers a compelling alternative with origins in the hardness of solving systems of polynomial equations. Unlike lattice-based schemes, founded on geometric intractability, multivariate systems are based on algebraic intractability—i.e., the difficulty of solving multivariate nonlinear equations over finite fields. The approach has the potential for shorter keys and faster computations but offers unusual challenges in standardization and in cryptanalysis.

The Hardness of Solving Multivariate Polynomial Equations

The basic idea behind multivariate cryptography [28] is the established fact that an NPhard problem in general, it is hard to solve a set of polynomial equations with multiple variables over a finite field.

$$f_1(x_1, x_2, \ldots, x_n) = y_1 \tag{.10}$$
$$f_2(x_1, x_2, \ldots, x_n) = y_2 \tag{.11}$$
$$\vdots \tag{.12}$$
$$f_m(x_1, x_2, \ldots, x_n) = y_m \tag{.13}$$

where each f_i is a polynomial and variables and coefficients are over a finite field $\mathbb{F}_q$. Solving for a solution vector $(x_1, x_2, \ldots, x_n)$ becomes infeasible computationally as the system increases in size and complexity, especially when the polynomials are quadratic or of higher degree. This intrinsic difficulty is the foundation of cryptographic security.

The security of multivariate cryptosystems relies on several key mathematical properties:

1. *Field Operations:* All operations are performed in a finite field $\mathbb{F}_q$, where q is generally selected as a small prime power to achieve optimal performance while maintaining security.
2. *Polynomial Degree:* The majority of practical implementations employ quadratic polynomials, since offer the best possible trade-off between security and computational performance.
3. *System Composition:* The public key is build by a composition of mappings $P = L_1 \circ F \circ L_2$, where:
   - L_1 and L_2 are invertible affine transformations that conceal the system's structure
   - F represents a well structured nonlinear central mapping
   - The composition $(\circ)$ of these components forms the entire cryptographic transformation.

Knowledge of the Jacobian determinant is essential for determining the invertibility—and therefore the security—of multivariate systems. The Jacobian matrix of a polynomial system contains vital information about local invertibility properties:

1. *For a mapping* $F : \mathbb{F}_q^n \to \mathbb{F}_q^n$, the Jacobian determinant determines if the mapping is locally invertible at a point.
2. The connection between local and global invertibility (the Jacobian Conjecture) is an open math problem.
3. This math uncertainty affects the way we design and analyze secure multivariate systems.

3.7 Explicit vs. Implicit Systems: Two Strategies for System Design

Multivariate cryptosystems can be divided into explicit or implicit systems in a broad sense.

1. *Explicit Systems:* These systems explicitly specify the polynomial equations employed in encryption or signing. The public key explicitly consists of the coefficients of these polynomials. The trapdoor (secret key) is usually a way to ”hide” the structure of these polynomials, making them easy to solve for someone who knows the secret, but difficult otherwise. A common method is to compose the core polynomial mapping with invertible affine transformations L_1 and L_2 that have a simple equation with simple operations that encrypt and decrypt.
2. *Implicit Systems:* Instead of directly giving polynomial equations for someone to use as a public key, these equations are more subtle. The public key consists of a more complex mapping with more parts to them, which consists of sets, parts, linear maps, etc to compute the decryption.

Both systems, with their approaches, have the same goals of computing and securing the public key, where a key consists of a polynomial problem.

3.8 Rainbow[2]: A Multivariate Solution to Post-Quantum Signatures

One of the most significant efforts to provide solutions to these architectural challenges for post-quantum cryptography, multivariate signature schemes have been shown to be especially promising candidates. Rainbow, despite recent exposures, showcases both the potential advantages and intrinsic complexities of the multivariate strategy. The signature scheme illustrates how meticulous balance between security needs and computational efficiency can be attained in practice, along with emphasizing the persisting difficulties in realizing strong post-quantum security. This section will first give an overview of the Rainbow signature scheme, its underlying principles, and its current security status, in the context of post-quantum cryptography as a whole.

Multivariate Signatures: Efficiency and Quantum Resistance

The underlying idea of multivariate cryptography is the computational intractability of solving systems of multivariate polynomial equations, i.e., over finite fields. The problem, while hard for classical computers, is believed to be quantum computer attack-resistant with current algorithms. This makes multivariate cryptography a serious candidate for post-quantum signatures.

Signature schemes, in general, are critical in digital communication. They allow senders to authenticate a message to receivers, so they could verify whether the

message was tampered or not. A well-designed signature needs to be easy to compute and to work, and so it would need to be easy to verify.

Rainbow seeks to do this by using multivariate equations with polynomial code to ensure data security. It tries to provide an efficient algorithm that is easy to compute, and that the security can be made better by adding more parts of the code. It also seeks to be a drop-in replacement to make future adoptions more easy, so that all code, even PGP, can be used without a new implementation to make new signatures.

Rainbow's Architecture: Layering for (Intended) Security

Rainbow[2] is a specific type of multivariate signature scheme called an Unbalanced Oil and Vinegar (UOV) scheme with multiple layers. Each layer partitions variables into "oil" (secret) and "vinegar" (public) sets, with equations of the form:

$$f_i = \sum \alpha_{\{jk\}} v_j v_k + \sum \beta_{\{jk\}} v_j o_k + \sum \gamma_j o_j + \delta \tag{.14}$$

where v_j: vinegar variables, o_k: oil variables.

The layered approach is the key innovation of Rainbow. By dividing the equations into bands and the variables into corresponding sets, the designers hoped to create a complex web that would be resistant to linearization attacks, where an attacker tries to create linear relationships between the variables and equations.

Inside Rainbow: The Code's Generation

The signing process involves multiple steps that start with calculating the seed to encrypt the code. In particular, the main method to generate it, has a lot of math:

1. *Key Exchange* - Key exchange (The first code used to encrypt the signature) is the process by which you create matrices and seeds to distribute between sender and public.
2. *Hashing* - All modern schemes needs hashing since hashing produces a fixed string output. This would help reduce the length of the code itself.
3. *Message Signing* - The message will include keys to ensure that all parties have the secure information for the decryption, along with the authentication process.

With everything said, when finished, the message is transferred through all channels, in which the code will be verified and set up with it. In addition, there is a set way to construct a message, which ensures that there are no vulnerabilities from any side of the code.

The security of Rainbow's key generation[2] process hinges critically on the use of a cryptographically secure pseudo-random number generator (PRNG). The seeds used to derive the system's matrices and vinegar variables must be generated with high entropy to prevent attackers from reconstructing the private key. If the PRNG

is predictable or biased, an adversary could reverse-engineer the seed values, deduce the structure of the multivariate equations, and ultimately forge signatures. This dependency on strong randomness underscores the importance of robust implementation practices in multivariate cryptography, where even theoretically secure designs can fail due to weaknesses in entropy sources.

With all the information transferred, it is now onto its encryption to be created.

3.9 GeMSS[30]: A Hidden Equation-Based Multivariate Signature Scheme

Multivariate cryptography continues to attract strong attention as a potential avenue for the development of signature schemes that are quantum-resistant. GeMSS (Generalised Method for finding Short Solutions), in particular, represents an effort to build a practical signature algorithm based on the Hidden Field Equations (HFE) problem. While aiming for advantages like small key sizes, GeMSS[30] also faces scrutiny regarding its security and resistance to known attacks.

The Foundation: Hidden Field Equations (HFE)

GeMSS was derived from a foundation known as the Hidden Field Equations (HFE) problem. The HFE problem first defines a polynomial f over a big ”hidden” field extension K of a small finite field k. The map is then converted by two maps, linear map and affine map (or matrix), to the polynomial rings on both fields. A simple construction works, in that the private function is simple to compute, but is encoded into a very hard to solve scheme.

GeMSS Implementation Structure

The implementation of GeMSS[30] builds upon the HFE foundation through several key components:

1. *Private function (f):* A multivariate problem created using specific HFE parameter settings.
2. *Two affine transformations:* Used to mix the transformations and generate public keys.
3. *Public key generation:* Creates ciphertext that appears as random elements.

With the basic map now established, there are ways that will cut down on complexity that must be dealt with. GeMSS employs a number of key optimization methods to contain computational complexity in its operations. The scheme takes

advantage of optimal tower field representations in its construction of field towers, which presents an efficient mechanism for conducting calculations at various levels of fields. This is complemented by the use of effective algorithms for polynomial evaluation over the extension field, thus lowering the computational cost of such ubiquitous operations. GeMSS[30] also uses advanced matrix operation handling for affine transformations, which are crucial for signature generation and verification. These combinations work synergistically to increase the real-world efficiency of the scheme without compromising its security aspects[5,19].

3.10 Isogeny-Based Cryptography[31,33]: Exploring the Landscape of Elliptic Curves[9]

The previous sections have considered two leading post-quantum cryptography approaches: lattice-based systems, which utilize geometric hardness, and multivariate systems, which utilize algebraic hardness. This section now addresses a third paradigm that arises from classical elliptic curve theory: isogeny-based cryptography. This system is a masterful blend of old mathematical concepts with post-quantum security needs. While lattice cryptography works within geometric structures and multivariate cryptography works with polynomial equations, isogeny-based cryptography enters the complex world of mappings between elliptic curves. These mappings, isogenies, provide cryptographic schemes with significantly smaller key sizes and are rooted in many decades of theoretical foundations in number theory and algebraic geometry.

Elliptic Curves: The Foundation[9]

An *elliptic curve* E over a field k is a smooth, projective algebraic curve of genus one with a specified point O. When working over a field k of characteristic neither 2 nor 3, an elliptic curve admits a Weierstrass equation of the form:

$$y^2 = x^3 + Ax + B \tag{.15}$$

where A and B are elements of the field k, and the discriminant $\Delta = -16(4A^3 + 27B^2)$ is non-zero, ensuring the curve is smooth. The set of solutions (x, y) to this equation, along with a special point O (the point at infinity), forms an abelian group. The group operation, denoted by "+", is geometrically defined by the "chord-and-tangent rule." Points on it can be added and subtracted to solve encryption problems.

What is an Isogeny?

An isogeny ϕ is a non-constant rational map between two elliptic curves E_1 and E_2 that preserves their group structure, denoted as $\phi : E_1 \rightarrow E_2$. The mathematical properties of an isogeny are quite specific: it must be a morphism of algebraic curves that maps the point at infinity of E_1 to the point at infinity of E_2 (that $\phi(O_{E_1}) = O_{E_2}$, and it must preserve the group operation, meaning that for any points P and Q on E_1, we have $\phi(P + Q) = \phi(P) + \phi(Q)$. The measure of an isogeny's complexity is given by its degree, denoted $\deg(\phi)$, which is defined as the degree of the function field extension $[k(E_1) : \phi^* k(E_2)]$. When this degree takes a specific value ℓ, we refer to the mapping as an ℓ-isogeny.

3.11 The SIKE Vulnerability: A Turning Point

In 2022, researchers Wouter Castryck and Thomas Decru disclosed a disastrous classical attack that effectively weakened SIKE's security[33]. The novelty of the attack was its utilization of the De Feo-Jao approach, which provided a solution to the problem of isogeny path generation. The principal mechanism of the attack operated through the determination of smooth degrees: if there is an L-degree isogeny between curve I and some curve J where there is an order-2 solution, then the encryption is vulnerable to successful cryptanalysis.

The attack's success relied on its ability to derive meaningful information about the isogeny path from the initial curve data. By leveraging mathematical properties of the underlying elliptic curve framework, attackers were able to recover the so-called secret isogeny paths. This vulnerability was particularly devastating since it demonstrated that the underlying mathematical postulates on which SIKE's security rested were faulty, as opposed to uncovering an implementation vulnerability.

Even though SIKE's breakdown was a huge setback to isogeny-based cryptography, the topic itself still possesses valuable properties worth exploring further. The strengths of the systems themselves - namely their compact key sizes and low-implementation complexity - continue to motivate research into alternative systems. The present research is directed towards discovering more secure mathematical underpinnings that would be resistant to both the De Feo-Jao type of attacks and potential quantum attacks.

4 Results and Discussions

The insistent progress of quantum computing forces a paradigm change in cryptographic infrastructures to protect contemporary structures. Quantum computers, which can run Shor's and Grover's algorithms, represent a profound threat to ubiquitous cryptographic infrastructures supporting secure communication, blockchain

systems, and IoT networks. Post-Quantum Cryptography (PQC) offers a necessary solution, relying on sophisticated mathematical problems resistant to quantum and classical attacks.

This part examines the real-world applications of PQC in areas of prime interest, such as blockchain, cryptocurrencies, secure communication, IoT, and government adoption. These applications place the critical importance of the transition to quantum-resistant systems alongside the trade-offs of deploying PQC solutions.

4.1 Blockchain[34] and Cryptocurrencies[40]: Fortifying the Foundation

Security of blockchain[1] largely depends on asymmetric cryptography for signatures and proof-of-work (PoW) mining. Quantum computing poses the risk to these cryptographic primitives, specifically:

- *Elliptic Curve Digital Signature Algorithm (ECDSA)* → applied to wallet authentication.
- *Proof-of-Work (PoW)* mining (SHA-256 in Bitcoin, Keccak in Ethereum 2.0) → susceptible to Grover's speedup.

Quantum Threats to Blockchain: Unpacking the Risks

ECDSA relies on the *Elliptic Curve Discrete Logarithm Problem (ECDLP):*

$$P = kG \tag{.16}$$

Where P represents the public key, k is the private key, and G is a generator point on the elliptic curve. On classical computers, recovering k from P is computationally infeasible. However, *Shor's Algorithm* makes it possible for quantum computers to compute this problem in polynomial time, potentially making it possible for attackers to extract private keys from public keys, facilitating theft of funds.

The practical implications are significant, potentially destabilizing trust in decentralized finance (DeFi) and requiring proactive cryptographic agility.

Practical Implication: If quantum computers become fault tolerant with ~2330 logical qubits by 2040, they could break 256-bit ECDSA, rendering Bitcoin wallets vulnerable. At current Bitcoin prices, a successful attack could lead to billions of dollars in stolen cryptocurrency.

Bitcoin's use of ECDSA means that, in some cases—especially when addresses are reused—a public key may be exposed during transactions, potentially increasing risk.

Furthermore, *Grover's Algorithm* accelerates brute-force attacks on PoW algorithms like SHA-256 by reducing the computational difficulty from $O(N)$ to $O(\sqrt{N})$.

This would disproportionately benefit quantum miners, centralizing blockchain mining power. While current quantum computers are not powerful enough to perform profitable mining, the future may change, threatening the decentralization that is so valued in Blockchain mining.

To summarize these vulnerabilities, the table below outlines Bitcoin's risk profile under quantum attack scenarios.

Table .4 Bitcoin's Risk Profile Under Quantum Attacks

Threat	Description	Potential Consequence
Wallet Theft	Bitcoin wallets use ECDSA signatures that expose public keys during transactions. Quantum attackers could derive private keys from exposed public keys.	Massive financial losses for wallet holders. Erosion of trust in Bitcoin technology. Reduced adoption and usage.
Mining Centralization	SHA-256 (Bitcoin PoW) requires brute-force nonce search. Grover's Algorithm reduces mining difficulty quadratically, enabling quantum miners to centralize processing power.	Increased 51% attack risk enabling transaction reversal and double-spending. Undermines decentralization principles. Loss of network confidence.
Code Vulnerabilities	Quantum computers might exploit vulnerabilities in smart contracts and blockchain implementations.	Smart contract exploitation and network disruption. Potential fund and data loss.

Quantum-Resistant Blockchains: The Solution

In response to these risks, several blockchain platforms are now experimenting with quantum-resistant approaches. Alongside these innovations, secure communication protocols are also undergoing changes to counter quantum threats.

Quantum-Resistant Ledger (QRL): QRL is the first blockchain natively designed for quantum resistance. It employs XMSS (Extended Merkle Signature Scheme), a hash-based signature algorithm that is provably secure against quantum attacks. While XMSS offers robust security, its larger signature sizes—around 2 KB—demand more storage and bandwidth, which could be a concern for systems with limited resources.

IOTA's WOTS+ Implementation: IOTA has implemented Winternitz One-Time Signatures (WOTS+)[38], in place of ECDSA. This is quantum-resistant but necessitates one-time key use per transaction, thus being less efficient than conventional schemes. Nevertheless, WOTS+ brings trade-offs, specifically reduced transaction throughput, affecting overall network scalability.

Table .5 Case Study: QRL's XMSS Transactions

Metric	Value	Implication
Signature Size	~2 KB per transaction	Increased storage costs for all nodes. Higher bandwidth requirements.
Security Level	Provably quantum-resistant	High confidence in long-term security against quantum attacks.
Integration Complexity	Requires fundamental changes to blockchain protocol	Significant development effort and potential compatibility issues.
Performance Overhead	Slightly slower than ECDSA systems	May impact scalability in high-throughput scenarios.

4.2 Secure Communication: Guarding Against Eavesdropping

TLS, PGP, and secure messaging protocols are based on RSA, ECC, and DH, which are susceptible to quantum attacks. Post-quantum cryptography provides long-term confidentiality and forward secrecy.

Securing Email and Messaging

TLS, PGP, and Signal secure communication protocols are based on RSA, ECC, and Diffie-Hellman (DH), which are all susceptible to quantum attacks. Post-quantum cryptographic alternatives give a solution to make these protocols secure against quantum attackers in the future.

The migration is also made more complex by the necessity of backward compatibility, with hybrid solutions combining classical and post-quantum algorithms.

Protocols such as PGP employ RSA or ECC for public-key encryption. With the advent of quantum computers running Shor's Algorithm, messages can be decrypted in an instant. In response, email services such as ProtonMail are already testing hybrid encryption schemes that pair classical cryptography (e.g., AES-GCM) with lattice-based KEMs such as Kyber.

Conversely, messaging services such as WhatsApp and Signal are considering hybrid PQC models. Google's "BoringSSL" already supports Kyber-based KEMs for testing postquantum secure messaging. These works prove that PQC can maintain long-term confidentiality and forward secrecy in communication systems.

PQC in TLS/SSL Protocols

TLS underlies secure web communication but depends on ECDH and RSA for key exchange. Post-quantum TLS handshakes support lattice-based KEMs such as Kyber to provide security against quantum attackers.

Table .6 PQC-Enhanced TLS Handshake Process

ragged2e

Step	**Description**	**Security Implication**	**Performance Consideration**
Client Hello	Requests Kyber-based key exchange	Initiates quantum-resistant negotiation.	No significant overhead.
Server Hello	Responds with a quantum-safe public key.	Establishes a foundation for a quantum-safe connection.	Increased key size might marginally increase bandwidth.
Client Key Exchange	Generates a shared quantum-resistant session key.	Establishes a secure, quantum-safe shared secret.	Higher computational cost vs traditional handshakes.
Handshake Completion	Establishes a secure, quantum-safe connection.	Ensures confidentiality and integrity against classical and quantum attacks.	Implementation-dependent computational costs.

In addition to protecting emails and web traffic, post-quantum cryptography is also confronted with special challenges in the field of IoT and edge devices, where energy efficiency and computational power are paramount.

4.3 IoT and Edge Devices

Low-power, efficient cryptography is needed in IoT devices, which makes adoption of PQC difficult. Low-computational-resource environments like IoT sensors, smart meters, and embedded systems have special challenges in implementing PQC because they have limited energy budgets, memory, and computational power.

Adding to these are the distributed nature of IoT networks, requiring effective key management and over-the-air (OTA) updates.

Lightweight PQC Algorithms

A number of PQC algorithms have been identified as promising candidates for IoT and edge devices, striking a balance between security, efficiency, and resource limitation. These are Kyber, SABER, and FrodoKEM.

A comparison of these algorithms is given below.

Table .7 Lightweight PQC Algorithms for IoT

Algorithm	Key Exchange/ Signature	Security Level	Relative Key Size	Relative Computational Overhead	Notes
Kyber	Key Exchange	NIST L3	Medium	Medium	Lattice-based, offers good performance.
SABER	Key Exchange	NIST L3	Small	Low	Lattice-based, optimized for low-power devices.
FrodoKEM	Key Exchange	NIST L3	Large	High	Lattice-based, relies on different assumptions (LWE) with larger key sizes and slow operation.

The table shows a comparison of several algorithms, highlighting key size and computation overhead. Though the labels 'small' or 'large' give a general guide, the accompanying notes clarify how such variables could impact actual-world performance measures, including battery life and processing speed.

Implementation Challenges

Incorporating PQC into IoT and edge devices is not without challenge, mainly in the domain of power usage, memory consumption, and computational cost.

1. *Energy Utilization:*
   - Lattice-based cryptographic techniques, though secure, involve more computational complexity than conventional ECC, resulting in increased power consumption.
   - ARM Cortex-M processor optimizations and hardware acceleration are essential to make PQC viable for IoT devices.
2. *Memory Usage:*
   - McEliece-type schemes have huge public keys (up to 1 MB), which will outstrip the memory capacity of most IoT devices.
   - Kyber has a better balance between security aspects and intrinsic dimensions, therefore it is a more practical choice for constrained environments.

Other than consumer and industrial applications, the government and defense industries are also affected by pressure to implement quantum-resistant measures—considering the high stakes of protecting classified data.

4.4 Defense and Government

Governments must enforce PQC in order to secure classified data and key infrastructure against Future quantum threats. National security depends on quantum-resistant cryptography to provide that sensitive information will be secure even with the rise of quantum computers.

Protecting Classified Data

The NSA's Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) requires a shift to post-quantum cryptography by 2030. Quantum-resistant algorithms like Kyber and Dilithium are being used for secure government communications and data storage. These steps ensure that classified information remains safe from future quantum attackers.

Global Government PQC Initiatives

Multiple nations are actively pursuing PQC adoption and standardization. Major initiatives include:

Table .8 Global Government PQC Initiatives

Country	PQC Initiative
USA (NSA)	CNSA 2.0: Kyber, Dilithium for Federal use.
EU	Migration of government systems to PQC.
China	Hybrid Quantum Key Distribution (QKD) + PQC via Micius satellite.

These global efforts underscore the paramount importance of PQC in achieving geopolitical stability and data sovereignty.

4.5 Mapping the Post-Quantum World: Challenges and Prospective Directions

The quantum computing potential is vast when applied to cryptography. In 2019, *Google's Sycamore* processor claimed quantum supremacy by executing a complex calculation in mere seconds that would take millennia for a standard supercomputer. This feat highlights a vital fact: quantum computing really can threaten the safety

of our web world. *IBM's Osprey processor (2022)*, which contains 1,121 qubits, further demonstrates quick progress. Industry roadmaps foresee cryptographically meaningful quantum computers – computers that can factor RSA and ECC – to be available between 2030 and 2040, the National The *National Security Agency (NSA)* has started a large-scale migration to Post-Quantum Cryptography. (PQC) 2030 for sensitive communications via its Commercial National Security Algorithm Suite 2.0 (CNSA 2.0). The question is not whether but how we prepare.

This part addresses the technical and operational consequences of PQC transition, such as performance constraints, backward compatibility requirements, and the promise of the hybrid cryptographic techniques. We will also identify excellent research questions and evaluate breakthrough technologies that characterize the future of cryptography, such as zero-knowledge proofs, homomorphic encryption, and quantum-resistant blockchain technologies. Eventually, success in this transformation is not just reliant on technological advancement but also on vision, active collaboration, together with an active awareness of the trade-offs.

4.6 Implementation Challenges: The Devil is in the Details

Migration to PQC is not an effortless one; it possesses serious implementation issues that must be addressed proactively.

Performance Overhead: A Fine Balance

The most evident obstacle to bulk PQC adoption is the performance overhead related to quantum-resistant algorithms. While classical cryptographic algorithms such as RSA and ECC have been optimized for decades, most of the PQC candidates require much larger keys, larger dimensions and higher computational power. This translates to longer encryption/decryption times, increased memory usage, and higher bandwidth consumption.

Consider this: Kyber-768, a leading lattice-based key encapsulation mechanism (KEM), uses a public key of 1,184 bytes and a ciphertext of 1,088 bytes. By stark contrast, ECDH-256 requires only 32 bytes for a public key – Kyber is almost 37 times larger! Similarly, Dilithium-2, a lattice-based digital signature scheme, produces 2.4 KB signatures, while ECDSA-256 signatures are a mere 64 bytes.

Performance benchmarks demonstrate that TLS 1.3 handshakes using Kyber-768 experience a 25-30% increase in handshake latency compared to traditional ECDH-based key exchanges. While modern cloud computing environments may absorb this cost, resource-constrained devices – such as IoT sensors, implantable medical devices, and smartcards – may struggle significantly with the increased computational load. What was once negligible now becomes a bottleneck.

Addressing the Issue: Researchers are exploring various mitigation strategies. Hardware acceleration methods, including Field-Programmable Gate Arrays (FPGAs) and Application-Specific Integrated Circuits (ASICs), are under study to accelerate lattice cryptography implementations. Cryptographic optimizations – including structured lattices and batched polynomial multiplications – are also promising areas for enhancing efficiency. But the fundamental challenge persists: how to reconcile ironclad security with computational practicality in the real world. There is no magic bullet.

These future improvements are likely to demand continued investment in research and development to ensure they serve the varied needs of various computational environments.

Backward Compatibility: Bridging the Divide

The shift to PQC is not a software patch. Dozens of cryptographic systems – from financial networks and military communications to industrial control systems – were developed without consideration of quantum threats. Upgrading those systems requires careful modifications to cryptographic libraries, network protocols, and authentication infrastructures. Most importantly, PQC has to integrate seamlessly with legacy cryptographic standards.

The Long-Term Signature Problem: One of the most urgent concerns concerns long-term digital signatures. Take today's digital certificates, for example – like X.509 certificates used for Transport Layer Security (TLS) and digital identities – these have to be secure for decades. A future quantum attacker capable of compromising these signatures will then be able to retroactively forge or sabotage past transactions and stored data. Government records, legal contracts, and blockchain transactions (in which data integrity has to be maintained indefinitely) are at highest risk. This is a time bomb ticking away.

This problem highlights the requirement for cryptographic agility and proactive actions to ensure longterm data integrity against future threats. Governments and companies both have to take into account the implications regarding legal and regulatory compliance, intellectual property protection.

Cryptographic Agility as a Solution: Financial institutions and cloud service providers are emerging as advocates of cryptographic agility – a strategic strategy that makes systems able dynamically alternate between classical and quantum-resistant cryptography algorithms. This enables a staged move, while business continues without the interference of pre-existing security frameworks. But agility in cryptography is not enough; hybrid encryption tools combining both classical and quantum-resistant methods need to be deployed to achieve layered security.

4.7 Hybrid Approaches: A Symphony of Security

Hybrid cryptographic systems represent a pragmatic means of migrating towards PQC while not leaving traditional cryptographic infrastructures behind. They intentionally pair a classical cryptographic scheme (e.g., ECDH, RSA) with a post-quantum scheme (e.g., Kyber, Dilithium) – resulting in strong resilience against both classical and quantum attacks.

The Hybrid TLS 1.3 Handshake (Kyber + ECDH): A Step-by-Step Example

- *Client Hello:* The client suggests a hybrid key exchange (Kyber-768 + ECDH-256).
- *Server Hello:* The server chooses the hybrid suite and sends both Kyber and ECDH public keys.
- *Key Exchange:* Both client and server each compute two shared secrets: Kyber shared key (quantum-secure), and ECDH shared key (classical security).
- *Session Key Derivation:* Both the keys are merged by a hash-based key derivation function (HKDF) to create a final encryption key.

Top tech companies, including Google and Cloudflare, have successfully experimented with Kyber + ECDH hybrid encryption for TLS, where only a 25% increase in handshake time was demonstrated by performance benchmarks, is a viable bridging solution.

Key Note: Hybrid encryption is effective but at the cost of increased computational complexity and longer handshake messages requiring efficient key management techniques. Efficient key management solutions are required here to prevent exposing new vulnerabilities.

4.8 Research and Collaboration: Forging a Secure Future

Development and deployment of PQC are not individualistic endeavors; they require international collaboration and ongoing research efforts.

Global Efforts: A Worldwide Imperative

The National Institute of Standards and Technology (NIST) PQC Standardization Project is at the forefront of worldwide efforts to create and standardize post-quantum cryptographic algorithms. Concurrently, the National Security Agency (NSA)'s Commercial National Security Algorithm Suite 2.0 (CNSA 2.0) program requires PQC implementation for classified government communications by 2030. The European Union (EU) and China are also creating standalone post-quantum cryptographic architectures.

These advances will not only determine the future of secure systems but also the manner in which industries and consumers engage with digital technology, potentially revolutionizing trust in digital transactions and communications.

Open Problems in PQC: Uncharted Territory

Despite progress, some essential research challenges in PQC are yet to be resolved. These include:

- *Side-Channel Attacks on Lattice Cryptography:* Power analysis attacks, which exploit voltage fluctuations during encryption, can expose secret keys. Research into masked lattice encryption implementations is essential to mitigate these risks.
- *Optimizing PQC for IoT and Mobile Devices:* Some algorithms, such as FrodoKEM, offer high security but feature large keys, making them impractical for resource-constrained mobile devices. Lightweight alternatives, such as SABER and compressed lattice techniques, are being actively explored. Careful hardware and software co-design is required.
- *Post-Quantum Blockchain Security:* Classical blockchain signatures (e.g., ECDSA in Bitcoin) are vulnerable to Shor's Algorithm. Lattice-based zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs) and zero-knowledge scalable transparent arguments of knowledge (zk-STARKs) are being developed for quantum-resistant, privacy-preserving blockchain transactions. This remains an active area of research.

4.9 Future Trends: The Horizon Beckons

Beyond the immediate challenges of PQC deployment, several emerging trends hold the potential to transform the future of cryptography.

- *Zero-Knowledge Proofs (ZKPs):* ZKPs enable one party to prove to another that a statement is true without revealing any information beyond the veracity of the statement itself. In the post-quantum world, lattice-based ZKPs will be invaluable for privacy-preserving authentication and secure multi-party computation.
- *Fully Homomorphic Encryption (FHE):* FHE allows computations to be performed on encrypted data without requiring decryption. This technology could revolutionize secure cloud computing by enabling organizations to process sensitive data without ever exposing it to the cloud provider.
- *Quantum Key Distribution (QKD):* QKD uses the principles of quantum mechanics to securely distribute encryption keys. While QKD is not a post-quantum cryptographic algorithm per se, it offers a complementary approach to securing communications channels. However, QKD faces practical limitations in terms of distance, cost, and infrastructure requirements.

Long-Term Vision: A Quantum-Resistant Future

The long-term vision is to establish a completely quantum-resistant digital foundation by 2040. Governments, banks, cloud providers, and standards organizations need to coordinate actively to facilitate a smooth and gradual transition before quantum computers are strong enough to break today's cryptographic underpinnings. What we do today will determine the security of our digital future for decades. PQC is not merely an upgrade in technology; it is a paradigm shift at its very core cybersecurity. To successfully make this transition requires concerted effort across industries, continued investment in research and development, and a dedication to creating a secure and quantum-resistant future. The moment to act is now. Based on the challenges and opportunities described above, a number of key actions are required for progressing:

1. *Recommendation 1:* Make cryptographic agility a priority across all systems. This will allow for quick adaptation to new cryptographic standards and new quantum threats. This needs not just technical ability but organizational processes and policies that facilitate agile cryptography.
2. *Recommendation 2:* Make investments in hardware acceleration and optimization methods for PQC algorithms. This is key to reducing the performance overhead and making practical usability in resource-poor settings like IoT devices and mobile apps possible.
3. *Recommendation 3:* Encourage more cooperation between governments, industry, and academia. Standardization of PQC algorithms, collaboration on threat intelligence, and best practices demand coordinated action on an international level. This can serve to promote interoperability and mitigate the risk of disjointed security strategies.

5 Conclusion: From Threat to Transformation – Forging a Quantum-Resilient Future

This chapter has not only described a threat; it has mapped a path for transformation. The unstoppable march of quantum computing forces a wholesale rethinking of our cryptographic roots. Although the precise timing is unclear, the direction is evident: existing cryptographic standards are due to become obsolete, requiring proactive and sweeping measures. The shift towards PQC is imminent and inevitable. In order to achieve long-term security in the quantum age, it will be crucial for networks such as Bitcoin and Ethereum to switch from ECDSA to quantum-resistant substitutes like XMSS or Dilithium. Bitcoin and Ethereum need to replace ECDSA using quantum-resistant algorithms such as XMSS or Dilithium. TLS 1.3 should implement Kyber-based key exchange protocols, and IoT devices should incorporate lightweight algorithms such as SABER or FrodoKEM. Governments across the globe are leading the implementation of PQC, guaranteeing the security of essential

infrastructures. PQC is not an enhancement—it is the future of secure cryptographic systems. Notwithstanding the standardization by NIST and early deployments of hybrid systems, there are still significant challenges. The performance overhead of PQC algorithms, the intricacies of text-back compatibility, and the ever-present danger of newly found vulnerabilities require continuing guard and innovation. Those challenges are not, however, to be considered insurmountable barriers but opportunities for strategic progress.

The future course needs a multi-faceted solution:

1. *Cryptographic Agility:* Design should support not only the currently existing PQC standards while simultaneously accommodating forthcoming advancements in cryptography. This necessitates a modular architectural frameworks, standard protocols, and an active security management culture.
2. *Hybrid Deployments:* The intentional combination of legacy and post-quantum algorithms offers a practical approach to limiting risk during the transition period. This approach requires serious scrutiny of key management challenges, performance compromises, and the chance for developing flaws.
3. *Sustained Research & Development:* Allocation of resources towards evolving cryptographic models – such e.g., zero-knowledge proofs, homomorphic encryption, and quantum key distribution – is Critical to long-term security and gaining a competitive advantage. This entails strict cryptanalysis of current PQC candidates and research into new mathematical Foundations.
4. *Global Collaboration:* Industry, government, and academia have to cooperate to produce Defining clear criteria, sharing threat intelligence, and coordinating deployment efforts. An uncoordinated such a plan will only raise the chances of overall security breakdowns.

The shift to PQC is not merely a technical necessity; it is a strategic chance to enhance the safety, strength, and reliability of our digital systems. Through the adoption innovation, fostering co-operation, and positively addressing the challenges of the future, we can build a world in which safe communication and protection of data are not threatened by the introduction of Quantum computing requires not just a reaction to the threat, but we must be proactive in creating what is required to further the field.

Acknowledgements The authors would like to thank the cryptography research community for their tireless efforts in developing post-quantum cryptographic solutions and their contributions to the standardization process. We also express our gratitude to the National Institute of Standards and Technology (NIST) for their leadership in the PQC standardization initiative. Special thanks to Department of Data Science and Bushiness System, SRMIST for supporting this research work.

Competing Interests The authors have no conflicts of interest to declare that are relevant to the content of this chapter.

Ethics Approval This research work did not involve any human participants or animals, and therefore no ethics approval was required.

Appendix: Glossary of Terms

Asymptotic Complexity — A mathematical notation that describes the runtime or space requirements of an algorithm as the input size grows. Common notations include O(n) (linear), $O(n^2)$ (quadratic), and $O(2^n)$ (exponential).

Elliptic Curve Cryptography (ECC) — A public-key cryptography approach based on the algebraic structure of elliptic curves over finite fields. ECC allows smaller keys compared to non-EC cryptography to provide equivalent security.

Key Encapsulation Mechanism (KEM) — A public-key encryption technique used to securely exchange symmetric encryption keys. KEM is designed to be secure against chosen-ciphertext attacks.

Learning With Errors (LWE) — A computational problem where the task is to recover a secret vector s given a matrix A and a vector b = As + e, where e is a small error vector. LWE is believed to be hard even for quantum computers.

Post-Quantum Cryptography (PQC) — Cryptographic algorithms that are believed to be secure against attacks from both classical and quantum computers. PQC aims to replace vulnerable algorithms like RSA and ECC.

Quantum Fourier Transform (QFT) — A quantum algorithm that performs a change of basis on quantum bits, analogous to the classical discrete Fourier transform. QFT is a key component of Shor's algorithm.

Shor's Algorithm — A quantum algorithm for integer factorization that runs in polynomial time, threatening the security of RSA and other public-key cryptosystems based on the hardness of factoring.

References

1. Aggarwal, D., Brennen, G., Calvo, F., Chapdelaine, M., Mosca, M., & Paniagua, C. (2018). Quantum attacks on Bitcoin, and how to protect against them. *Designs, Codes and Cryptography*, 86(5), 1017–1039.
2. Alagic, D., Alperin-Sheriff, J., Apon, D. A., Cooper, D., Dang, Q. T., Kelsey, J., . . . & Zhang, Y. (2019). Status report on the second round of the NIST post-quantum cryptography standardization process. *NISTIR*, 8240.
3. Bernstein, D. J., Buchmann, J., & Dahmen, E. (Eds.). (2009). *Post-quantum cryptography*. Springer Science & Business Media.
4. Bos, J. W., Costello, C., Ducas, L., Melià-Seguí, J., & Naehrig, M. (2018). Post-quantum key exchange for the internet and beyond. *Proceedings of the IEEE*, 106(7), 1103–1118.
5. Castryck, W., & Decru, B. (2022). An efficient key recovery attack on SIDH with practical parameters. In *Advances in Cryptology–CRYPTO 2022: 42nd Annual International Cryptol-*

ogy Conference, Santa Barbara, CA, USA, August 15–19, 2022, Proceedings, Part III (pp. 587–618).
6. Ekert, A., & Jozsa, R. (1996). Quantum computation and Shor's factoring algorithm. *Reviews of Modern Physics*, 68(3), 733.
7. Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177), 203–209.
8. Lloyd, S. (2000). Ultimate physical limits to computation. *Nature*, 406(6799), 1047–1054.
9. Miller, V. S. (1986). Use of elliptic curves in cryptography. In *Advances in Cryptology—CRYPTO'85 Proceedings* (pp. 417–426).
10. Mosca, M. (2018). Cybersecurity in an era with quantum computers: will we be ready? *IEEE Security & Privacy*, 16(5), 38–41.
11. Naehrig, M., Schwabe, P., & Vercauteren, F. (2011). New software speed records for cryptographic pairings. In *Advances in Cryptology—ASIACRYPT 2011* (pp. 1–18). Springer.
12. NIST. (2024). Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. *NISTIR 8413*.
13. NSA. (2015). Commercial National Security Algorithm Suite. *CNSS Advisory Memorandum 2015-02*.
14. Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120–126.
15. Shor, P. W. (1994). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science* (pp. 124–134). IEEE.
16. Ajtai, M. (1996). Generating hard instances of lattice problems. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (pp. 99–108). https://doi.org/10.1145/237814.237838
17. Bernstein, D. J., Lange, T., & Peters, C. (2008). Attacking and defending the McEliece cryptosystem. In *Post-Quantum Cryptography* (pp. 31–46). Springer.
18. Bernstein, D. J., Hopwood, D., Hülsing, A., Lange, T., Rijneveld, J., & Schwabe, P. (2015). SPHINCS+: Stateless Hash-Based Signatures. In *Advances in Cryptology – EUROCRYPT 2015* (pp. 368–397). Springer.
19. Castryck, W., & Decru, B. (2022). An efficient key recovery attack on SIDH with practical parameters. In *Advances in Cryptology – CRYPTO 2022* (pp. 587–618). Springer.
20. Ducas, L., Gama, N., & Vercauteren, F. (2018). Short Lattice Vector Enumeration and Sieving Algorithms. In *Advances in Cryptology – ASIACRYPT 2018* (pp. 651–681). Springer.
21. National Institute of Standards and Technology (NIST). (2023). Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. *NIST Interagency or Internal Report (NISTIR) 8413*.
22. Regev, O. (2005). On Lattice-Based Cryptography. *Journal of the ACM*, 52(6), 844–891.
23. Micciancio, D., & Goldwasser, S. (2002). *Complexity of Lattice Problems: A Cryptographic Perspective*. Kluwer Academic Publishers.
24. Regev, O. (2009). On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM*, 56(6), 1–40.
25. Bos, J. W., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., . . . & Stehlé, D. (2018). CRYSTALS – Kyber: a CCA-secure module-lattice-based KEM. In *Advances in Cryptology – EUROCRYPT 2018*.
26. Lyubashevsky, V., Peikert, C., & Regev, O. (2010). On ideal lattices and learning with errors over rings. In *Advances in Cryptology – EUROCRYPT 2010*.
27. Ducas, L., Lyubashevsky, V., & Schwabe, P. (2018). CRYSTALS – Dilithium: a lattice-based digital signature. In *Advances in Cryptology – ASIACRYPT 2018*.
28. Ding, J., Gower, J. E., & Schmidt, D. S. (2006). *Multivariate Public Key Cryptosystems*. Springer.
29. Ding, J., & Schmidt, D. (2005). Rainbow, a new multivariate polynomial signature scheme. In *Applied Cryptography and Network Security*.
30. Petzoldt, A., Bulygin, S., Buchmann, J., & Tarakanov, O. (2011). GeMSS–a signature scheme based on the MinRank problem. *Journal of Mathematical Cryptology*.

31. Jao, D., & De Feo, L. (2011). Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In *Advances in Cryptology – ASIACRYPT 2011*.
32. Charles, D., Goren, E. Z., & Lauter, K. (2009). Hash functions from expanding graphs. *Journal of Cryptology*.
33. Castryck, W., & Decru, T. (2022). An efficient key recovery attack on SIDH. In *Advances in Cryptology – EUROCRYPT 2022*.
34. Albrecht, M. R., et al. (2020). Estimating the cost of generic quantum pre-image attacks on SHA-2 and BLAKE2. *Designs, Codes and Cryptography*, 88(1), 17–42.
35. Barker, E., et al. (2016). Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography. *NIST Special Publication 800-56A, Revision 2*.
36. Bernstein, D. J., Buchmann, J., & Dahmen, E. (Eds.). (2009). *Post-quantum cryptography*. Springer.
37. Bernstein, D. J., et al. (2015). SPHINCS: practical stateless hash-based signatures. In *Advances in Cryptology—EUROCRYPT 2015* (pp. 368–397). Springer.
38. Castagnos-Hayes, A., Phan, R. C. W., & Pointcheval, D. (2019). Winternitz one-time signatures revisited. *Journal of Cryptographic Engineering*, 9(3), 239–253.
39. Chou, T. (2020). FrodoKEM: Learning with errors key encapsulation. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(3), 178–204.
40. Katz, J., & Lindell, Y. (2020). *Introduction to modern cryptography*. Chapman and Hall/CRC.
41. National Institute of Standards and Technology (NIST). (2022). Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process. *NISTIR 8413*.
42. National Security Agency (NSA). (2024). Commercial National Security Algorithm Suite 2.0.
43. Ogren, M., & Schanck, A. B. (2021). Saber: Module-LWR based KEM. *Journal of Cryptographic Engineering*, 11(3), 295–310.
44. Bos, J. W., Costello, C., Ducas, L., Melià-Seguí, J., & Naehrig, M. (2018). Post-quantum key exchange for the internet and beyond. *Proceedings of the IEEE*, 106(7), 1103–1118.
45. Alagic, D., Alperin-Sheriff, J., Apon, D. A., Cooper, D., Dang, Q. T., Kelsey, J., . . . & Zhang, Y. (2019). Status report on the second round of the NIST post-quantum cryptography standardization process. *NISTIR*, 8240.
46. Huland, S., et al. (2020). Performance analysis of post-quantum key exchange on ARM Cortex-M4. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2020(4), 452–477.